REPORT DOCUMENTATION PAGE

Form Approved OMB NO. 0704-0188

1. REPORT DATE (DD-MM-YYYY)	2. REPORT TYPE	3. DATES COVERED (From - To)
17-08-2015	Final Report	1-Feb-2012 - 31-Jan-2015

4. TITLE AND SUBTITLE

Final Report: Heterogeneous VM Replication: A New Approach to Intrusion Detection, Active Response, and Recovery in Cloud Data Centers

5a. CONTRACT NUMBER
W911NF-12-1-0055

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
206022

6. AUTHORS

Mohan Malkani, Sachin Shetty, Peng Liu

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES AND ADDRESSES

Tennessee State University
Research and Sponsored Program
3500 John A. Merritt Blvd.
Nashville, TN 37209 -1561

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS (ES)

U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSOR/MONITOR'S ACRONYM(S)
ARO

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
60432-CS-REP.5

12. DISTRIBUTION AVAILIBILITY STATEMENT

Approved for Public Release; Distribution Unlimited

13. SUPPLEMENTARY NOTES

The views, opinions and/or findings contained in this report are those of the author(s) and should not contrued as an official Department of the Army position, policy or decision, unless so designated by other documentation.

14. ABSTRACT

The goal of this program is to enable development of novel security methods to support future Air Force and Homeland Security in Cybersecurity enterprise. Developing the understanding and tools to build inherently secure software and to ensure the security of vast amounts of information flowing through relevant networks and information spaces are very germane to Air Force. One of the goals of AFOSR in information operations and security is to conduct research to develop new approaches to detection on intrusion, forensics, and active response and recovery from an attack on information systems. Tennessee State University is submitting a proposal to

15. SUBJECT TERMS

VM Migration, Moving Target Defense, Cloud Security

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	15. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Mohan Malkani
UU	UU	UU	UU		19b. TELEPHONE NUMBER 615-963-5400

Report Title

Final Report: Heterogeneous VM Replication: A New Approach to Intrusion Detection, Active Response, and Recovery in Cloud Data Centers

ABSTRACT

The goal of this program is to enable development of novel security methods to support future Air Force and Homeland Security in Cybersecurity enterprise. Developing the understanding and tools to build inherently secure software and to ensure the security of vast amounts of information flowing through relevant networks and information spaces are very germane to Air Force. One of the goals of AFOSR in information operations and security is to conduct research to develop new approaches to detection on intrusion, forensics, and active response and recovery from an attack on information systems. Tennessee State University is submitting a proposal to conduct research in developing H-VM-R (Heterogeneous VM Replication), a new approach to intrusion detection, active response, and recovery on servers in cloud data centers. Homogeneous VM replication is the state-of-the-art VM replication technology, but due to lack of artificial diversity, it is very limited in doing intrusion detection and active response. In contrast, H-VM-R does cost-effective intrusion detection by comparing heterogeneous VM images resulted from the same execution history, and cost-effective active response by proactively setting up standby VM replicas: migration from a compromised VM replica to a clean yet heterogeneous. VM replica is in fact the desired hot-start recovery. Our H-V-M-R research will address the specific USAF Cloud Computing requirements, such as scalable security monitoring, accountability, multi-abstraction isolation, security consolidation and elasticity.

Enter List of papers submitted or published that acknowledge ARO support from the start of the project to the date of this printing. List the papers, including journal references, in the following categories:

(a) Papers published in peer-reviewed journals (N/A for none)

Received Paper

TOTAL:

Number of Papers published in peer-reviewed journals:

(b) Papers published in non-peer-reviewed journals (N/A for none)

Received Paper

TOTAL:

Number of Papers published in non peer-reviewed journals:

(c) Presentations

Number of Presentations: 0.00

Non Peer-Reviewed Conference Proceeding publications (other than abstracts):

Received	Paper

TOTAL:

Number of Non Peer-Reviewed Conference Proceeding publications (other than abstracts):

Peer-Reviewed Conference Proceeding publications (other than abstracts):

Received	Paper	
08/17/2015	6.00	Xuebiao Yuchi, Sachin Shetty. Enabling Security-Aware Virtual Machine Placement in IaaS Clouds, Milcom. 27-OCT-15, . : ,
08/17/2015	7.00	Min Li, Zili Zha, Wanyu Zang, Meng Yu, Peng Liu, Kun Bai. Detangling Resource Management Functions from the TCB in Privacy-Preserving Virtualization, 19th European Symposium on Research in Computer Security (ESORICS 2014). 07-SEP-14, . : ,
08/19/2014	2.00	Ping Chen, Peng Liu, Bing Mao, Rui Wu. System Call Redirection: A Practical Approach to Meeting Real-world VirtualMachine Intropseciton Needs, Dependable Systems and Networks. 23-JUN-14, . : ,
08/19/2014	3.00	Sachin Shetty, Peng Liu, Jiwu Jing, Lingchen Zhang. RootkitDet: Practical End-to-End Defenseagainst Kernel Rootkits in a Cloud Environment, European Symposium on Research in Computer Security. 09-SEP-14, . : ,
08/19/2014	4.00	Sachin Shetty, Hellen Maziku. Towards a Network Aware VM Migration:Evaluating the Cost of VM Migration in CloudData Centers, IEEE Conference on Cloud Networking. 08-OCT-14, . : ,
08/30/2012	1.00	Shengzhi Zhang, Peng Liu. Assessing the Trustworthiness of Drivers, RAID 2012. 09-SEP-12, . : ,

TOTAL: **6**

Number of Peer-Reviewed Conference Proceeding publications (other than abstracts):

(d) Manuscripts

Received Paper

TOTAL:

Number of Manuscripts:

Books

Received Book

TOTAL:

Received Book Chapter

TOTAL:

Patents Submitted

Patents Awarded

Awards

Graduate Students

NAME	PERCENT_SUPPORTED	Discipline
Hellen Maziku	1.00	
Lingchen Zhang	1.00	
Biswajit Biswal	0.30	
FTE Equivalent:	**2.30**	
Total Number:	**3**	

Names of Post Doctorates

NAME	PERCENT_SUPPORTED
FTE Equivalent:	
Total Number:	

Names of Faculty Supported

NAME	PERCENT_SUPPORTED	National Academy Member
Sachin Shetty	0.30	
Peng Liu	0.30	
FTE Equivalent:	**0.60**	
Total Number:	**2**	

Names of Under Graduate students supported

NAME	PERCENT_SUPPORTED	Discipline
Dan Fishler	0.30	Electrical Engineering
Jordan Speller	0.30	Computer Science
Marc Primeau	0.50	Electrical Engineering
Jeremiah Cooper	0.50	Computer Science
FTE Equivalent:	**1.60**	
Total Number:	**4**	

Student Metrics

This section only applies to graduating undergraduates supported by this agreement in this reporting period

The number of undergraduates funded by this agreement who graduated during this period: 5.00

The number of undergraduates funded by this agreement who graduated during this period with a degree in science, mathematics, engineering, or technology fields:..... 5.00

The number of undergraduates funded by your agreement who graduated during this period and will continue to pursue a graduate or Ph.D. degree in science, mathematics, engineering, or technology fields:..... 3.00

Number of graduating undergraduates who achieved a 3.5 GPA to 4.0 (4.0 max scale):..... 5.00

Number of graduating undergraduates funded by a DoD funded Center of Excellence grant for Education, Research and Engineering:..... 0.00

The number of undergraduates funded by your agreement who graduated during this period and intend to work for the Department of Defense 3.00

The number of undergraduates funded by your agreement who graduated during this period and will receive scholarships or fellowships for further studies in science, mathematics, engineering or technology fields:..... 5.00

Names of Personnel receiving masters degrees

NAME
Pan Gao
Manu Misra
Prachi Mandil
Khurram Raza
Husam Adas
Total Number: 5

Names of personnel receiving PHDs

NAME
Total Number:

Names of other research staff

NAME	PERCENT_SUPPORTED
FTE Equivalent:	
Total Number:	

Sub Contractors (DD882)

1 a. Pennsylvania State University

1 b. Office of Sponsored Programs
110 Technology Center Building
University Park PA 168027000

Sub Contractor Numbers (c):
Patent Clause Number (d-1):
Patent Date (d-2):
Work Description (e):
Sub Contract Award Date (f-1):
Sub Contract Est Completion Date(f-2):

---

1 a. Pennsylvania State University

1 b. 110 Technology Center
200 Innovation Blvd.
University Park PA 168027000

Sub Contractor Numbers (c):
Patent Clause Number (d-1):
Patent Date (d-2):
Work Description (e):
Sub Contract Award Date (f-1):
Sub Contract Est Completion Date(f-2):

---

1 a. Pennsylvania State University

1 b. Applied Research Laboratory

State College PA 168040030

Sub Contractor Numbers (c):
Patent Clause Number (d-1):
Patent Date (d-2):
Work Description (e):
Sub Contract Award Date (f-1):
Sub Contract Est Completion Date(f-2):

Inventions (DD882)

Scientific Progress

* System Call Redirection: A Practical Approach to Meeting Real-world Virtual Machine Introspection Needs

Existing VMI techniques have high overhead, and require customized introspection programs/tools for different guest OS versions – lack of generality. In this work, we developed ShadowContext, a system for close-to- realtime manual-effort-free VMI. ShadowContext can meet several important real-world VMI needs which existing VMI techniques cannot. Compared to other automatic introspection tool generation techniques, ShadowContext has two merits: (1) Its overhead is signi¿cantly less. It achieves close-to-realtime VMI. (2) It signi¿cantly improves the practical usefulness of introspection tools by allowing one introspection program to inspect a variety of guest OS versions. These merits are achieved via a new concept called “Shadow Context” which allows the guest OSes system call code to be reused inside a “shadowed” portion of the context of the out-of-guest inspection program. Besides, ShadowContext is secure enough to defend against a variety of real world attacks. ShadowContext is designed, implemented and systematically evaluated. Experimental results show that the performance overhead is about 75% with a median initialization time of 0.117 milliseconds.

RootkitDet: Practical End-to-End Defense against Kernel Rootkits in a Cloud Environment”

In cloud environments, kernel-level rootkits still pose serious security threats to guest OSes. Existing defenses against kernel-level rootkit have limitations when applied to cloud environments. In this talk, we present RootkitDet, an end-to-end defense system capable of detecting and diagnosing rootkits in guest OSes with the intent to recover the system modifications caused by the rootkits in cloud environments. RootkitDet detects rootkits by identifying suspicious code region in the kernel space of guest OSes through the underneath hypervisor, performs diagnosis on the code of the detected rootkit to categorize it and identify modifications, and reverses the modifications if possible to eliminate the effect of rootkits. Our evaluation results show that the RootkitDet is effective on detection of kernel-level rootkits and recovery modifications with less than 1% performance overhead to the guest OSes and the computation and network overhead is linear with the quantity of the VM instances being monitored.

Enabling Security-Aware Virtual Machine Placement in IaaS Clouds

Infrastructure as a Service (IaaS) facilitates the provisioning of virtual machines (VMs) in cloud computing platform for disjoint customers in a highly scalable, flexible, and cost-efficient fashion. However, provisioning of new VMs should take into account presence of vulnerable co-resident VM. A vulnerable VM poses security risk to co-locating VMs and the physical machine. Thus, VM placement policies can have an impact on the overall security of the cloud computing platform. In this work, we quantify the security risks of cloud environments for VM placement schemes in presence of vulnerable VMs. Based on our security evaluation, we propose a novel virtual machine placement scheme that can minimize the security risks for the cloud platform. Experimental results demonstrate that our approach can improve the survivability of most virtual machines and reduce the threat of attacks in the cloud platform. The computing costs and deployment costs of our techniques are also practical.

Network Aware Resource Allocation in Cloud Data Center

Effective resource allocation algorithm is critical to ensure the performance of users’ applications as well as the efficiency of overall resource usage in cloud data center. We propose a new network-aware resource allocation solution based on minimum-height tree, with the goal of minimizing the maximum latency in communication between VMs as well as the overall network costs inside the data center. In our approach, we try to improve the effectiveness of resource allocation procedure by taking into account the hierarchical network topology characteristics of the data center. We also consider VM heterogeneities in terms of computational and communicational requirements to make our approach more practical. Simulations over exampled cloud systems imply that our approach can provide significant gains over other simpler resource allocation algorithms.

Technology Transfer

Heterogeneous VM Replication: A New Approach to Intrusion Detection, Active Response, and Recovery in Cloud Data Centers

Final Report

Abstract

The goal of the ARO grant W911NF-12-1-0055*,* was to develop new approaches to detection on intrusion, forensics, and active response and recovery from an attack on information systems. conduct research in developing H-VM-R (Heterogeneous VM Replication), a new approach to intrusion detection, active response, and recovery on servers in cloud data centers. Homogeneous VM replication is the state-of-the-art VM replication technology, but due to lack of artificial diversity, it is very limited in doing intrusion detection and active response. In contrast, H-VM-R does cost-effective intrusion detection by comparing heterogeneous VM images resulted from the same execution history, and cost-effective active response by proactively setting up standby VM replicas: migration from a compromised VM replica to a clean yet heterogeneous. VM replica is in fact the desired hot-start recovery. Our H-V-M-R research will address the specific USAF Cloud Computing requirements, such as scalable security monitoring, accountability, multi-abstraction isolation, security consolidation and elasticity. This report provides a summary of the technical approaches and accomplishments. **In summary, the project has resulted in 10 journals and conference publications. Two graduate students won awards for best presentation at Tennessee State University's annual research symposium. The project's results were also leveraged in securing external grants and contracts worth $4M from Boeing, DHS, NSF and AFRL. The grant played a critical role in TSU becoming a member of recently funded DoD Center of Excellence in Cybersecurity and DHS Center of Excellence in Critical Infrastructure Resilience.**

Introduction

In this project, a faculty and student team from Tennessee State University (TSU) and Pennsylvania State University (PSU) developed H-VM-R (Heterogeneous VM Replication), a new approach to intrusion detection, active response, and recovery on servers in cloud data centers. H-V-M-R addresses the specific USAF Cloud Computing requirements, such as scalable security monitoring, accountability, multi-abstraction isolation, security consolidation and elasticity. The *objectives* of H-VM-R approach are to:

- Make redundancy and high-availability practically affordable.
- Transform microscopic intrusion analysis and detection from pure offline security operations to an online capability directly participating in active response.
- Develop an innovative intrusion detection technology based on cross-VM inconsistency checking.
- Achieve fine-grained intrusion detection, response, and recovery.
- Develop a new artificial diversity technology which is simpler, more robust, and less expensive.

The team has developed several approaches to ensure H-VM-R provides an adequate intrusion detection and response. The research results have been published in 10 journals and conferences. In addition, two graduate students won prizes at TSU's annual research

symposium for their oral presentations of the research results. The project's results were also leveraged in securing external grants and contracts worth $4M from Boeing, DHS, NSF and AFRL. The grant played a critical role in TSU becoming a member of DoD Center of Excellence in Cybersecurity and DHS Center of Excellence in Critical Infrastructure Resilience. Below is a summary of the main technical approaches.

Approaches

1. *RootkitDet[7]*, an end-to-end defense to facilitate detection and recovery against known profile of kernel rootkits in a cloud environment
2. *Heterdevice[3]*, a novel device driver evaluation approach to comprehensively assess drivers against an implicit and complete model before putting any trust on them.
3. *ShadowContext[4]*, a system for close-to- real-time manual-effort-free VMI
4. Virtual machine placement scheme that can minimize the security risks for the cloud platform [1].
5. *MyCloud SEP[9]*, a novel architecture to separate resource allocation and management from the hypervisor in order to reduce the TCB size while supporting privacy protection.
6. Empirical evaluation of the network cost of moving VM during security attack[2,5].

Technical Approaches

RootkitDet: Practical End-to-End Defense against Kernel Rootkits in a Cloud Environment

Kernel-level rootkits are one of the most severe security threats in the operating systems. Although many softwares and research works have been devoted to detecting and preventing them, they still exist in the cloud environment because all of the softwares and research works focus on protection for a single operating system against rootkits, while the situation in the cloud is different to some extent. In cloud environment, the rootkits detection system should be efficient, scalable and easy to deploy. To achieve these objectives, we propose RootkitDet system to detect kernel-levelrootkits in the cloud environment.

In our design, the RootkitDet system [7] consists of one *conductor* and multiple *detector*s. The conductor runs on the host OS as a user space process. It communicates with all of the detectors through IPC. Basically, it sends detection commands to the detectors, and receives responses back. If rootkits are detected, it raises alert. The detector detects kernel-level rootkits in a VM by reading its registers and memory. In order to conveniently access the VM's registers and memory, the detector is integrated into the VMM which is called *qemu-kvm.*

We integrate the detector into qemu-kvm, which is the user-space tool of KVM. A VM based on KVM runs on the Host OS as a process. The detector is part of that process, so that it can easily access the state of the VM, including registers and memory. We initialize the detector after the creation of the VM instance. During the initialization, the detector establishes a connection to the conductor, and prepares to receive detection commands from the conductor. A detector performs three detection procedures. The first detection procedure requires the reconstruction of the list of loaded modules and the generation of the list of executable regions in the kernel space. For each executable region, the detector acquires its start address and size; for each module, the detector acquires its relocation address and the size of its memory region(which only depends on the size of its core executable code). Then the detector then examines the start address and size of each

executable region in the list, and finds out whether extra executable regions exist besides the regions of the kernel code and modules. The second detection procedure also requires the reconstruction of the list of loaded modules. The detector then determines the start address and size of unused space and checks whether some code resides in the unused space of each module. In order to determine whether any modification to the kernel and modules' code occurs in the last detection procedure, the detector calculates hash values for each regions of the kernel and modules' code, and compares them with original hash values, which are received from the conductor. The rootkits in the VM cannot cheat the detector by interfering with the generation of original hash values because they come from the conductor which is running on the host OS. In the first and second detection procedures, the detector reconstructs the module list to detect kernel-level rootkits and builds a description structure for each module, which contains the size of the core region as a property. However, the module list is built based on the VM's memory. That is to say, the description structures of modules are actually under the control of the rootkit if it is installed. In order to escape from the two detection procedures, rootkits may tamper with a module's size property of the core region. For example, the rootkit can modify a module's size property of the core region to a larger value. And it puts the its code right behind the module's core executable code, pretending itself as part of the module. Then it can escape from the first and second detection procedures. We leave this problem to the conductor and the conductor resolves it when generating the original hash values for all of the modules.

The conductor is a process running on a host OS as well as the VMs created by KVM. It accepts connections issued by detectors dynamically, and maintains those connections concurrently. The main function of the conductor is to decide when and which detector should detect kernel-level rootkits in a VM, and gives detection commands to it at a proper time. Basically, the conductor gives detection commands to a detector periodically. In fact, the conductor doesn't have to run on the same host OS as the detector because they communicates with each other through IPC. After sending detection commands to the detectors, the conductor waits for the responses from detectors. If the response represents that some rootkits are detected, the conductor raises an alert. The conductor is also responsible for generating original hash values of the loaded modules for each VM and sending them to the corresponding detector. The generation of original hash values should be indeed independent to the VMs so that rootkits in the VMs cannot interfere in. Therefore, the conductor needs to keep a copy of the original object file for each module, and then, does the same relocation work for a module as the kernel does when generating the original hash value. In order to keep a copy of each module's original object file, we require the registration for each module before it can be loaded by a VM. Cloud users are responsible for registering all of the modules that may be used by the VMs. During the registration of a module, its object file and name should be provided. The conductor records the registration and keeps a copy of the module's original object file.

The communication between the conductor and detector does not only include the detection commands coming from conductor to detector, but also includes initial data, auxiliary commands and response. The detector use the initial data to bridge the semantic gap between the raw data from VM's memory and data structures used by the VM's kernel. The purpose of the auxiliary commands is to help the conductor correctly generate original hash values of modules for each detector. RootkitDet system is scalable because the connections between detectors and the conductor are dynamically established and the conductor can manages multiple detectors.

Our evaluation results show that the RootkitDet system can detects all of the longterm kernel level rootkits, and the performance overhead is less than 1%. The complexity of the RootkitDet system is linear with the quantity of VM instances being monitored, and thus acceptable for scalability. These results highlights the promise of our system and indicate that the RootkitDet system is an adoptive choice to detect kernel-level rootkits in the cloud environment.

Assessing the Trustworthiness of Drivers and Detecting Malicious Driver Behavior through Heterogeneous VM Replication

A significant portion of the *attack surface* of (cloud) data centers is the driver code that runs inside each VM (Virtual Machine). A recent reality check study shows that over 70% percent of the Linux Operating System code-base is actually occupied by driver code. Based on this fact, attackers have shown stronger and stronger interests in compromising data centers through drivers. In fact, drivers have already become one of the weakest links of today's data centers.

Drivers, especially third party drivers, could contain malicious code (e.g., logic bombs) or carefully designed-in vulnerabilities. Generally, it is extremely difficult for static analysis to identify these code and vulnerabilities. Without knowing the exact triggers that cause the execution/exploitation of these code/vulnerabilities, dynamic taint analysis cannot help either.

Partially funded by this grant, we developed a novel driver evaluation approach, *Heterdevice*, to comprehensively assess drivers against an implicit and complete model before putting any trust on them. Heter-device [3] relies on virtual platforms to emulate heterogeneous device (Heter-device) pairs (e.g., Intel 82540EM NIC and Realtek RTL8139) for guest operating system replicas. Each replica loads heterogeneous drivers corresponding to the devices it runs on. Heter-device approach stands on the assumption that heterogeneous drivers should not have the same exploitable vulnerability due to their separated developing processes. So they provide an implicit and complete reference model for each other when trustworthiness assessment is conducted via fine-grained auditing. Hence, by deploying Heter-device as a high-interaction honeypot, we can closely compare the divergence of two replicas when the vulnerable driver is being compromised and leveraged.

The two replicas with heterogeneous drivers are synchronized at the exported function entry points, which are declared by OS kernel and implemented by each driver. We start a fine-grained auditing of driver's execution whenever kernel calls the corresponding driver functions. During driver's execution, every jump, call or return to kernel or other kernel modules' address space are logged for verification. The logs from heterogeneous drivers are parsed and compared to check any suspicious control flow redirection, e.g., one driver jumps to a kernel segment written by itself, while the other does not exhibit such behaviour. Moreover, any modification to key kernel data by drivers is recorded and verified against the heterogeneous drivers to check if it is a legitimate modification or a malicious manipulation.

We also deal with passive attacks launched from compromised drivers, e.g., network card driver intercepts incoming/outgoing packets and redirects them to remote entities. Thus, the network outgoing packets of the two replicas are audited and compared to find mismatch. Additional amount of traffic on one replica against the other suffices an alarm of confidentiality compromise.

Finally, abuse of kernel APIs, such as spin lock or kernel memory allocation requests, may cause CPU or memory starvation. Hence, any call to these resource request
APIs from drivers is also verified against heterogeneous drivers. By placing the synchronization and monitoring "sensors" in Heter-device, our honeypot can faithfully reveal multiple attack vectors of compromised drivers, including kernel integrity manipulation, resource starvation, and confidentiality tampering.

Compared to other diversity-based intrusion detection approaches such as N-variant, Heterdevice is the first work that does systematic in-depth modeling and analysis of the fine-grained interactions between drivers and the core kernel. For example, during driver's execution, Heterdevice is the first work that logs every jump, call or return to kernel or other kernel modules' address space. This enables Heterdevice to observe and analyze driver behavior at a much finer-grained level than existing approaches. This is why Heterdevice can assess the trustworthiness of drivers while other approaches could not.

We have designed and fully implemented the Heterdevice system prototype. Evaluation shows that this approach can faithfully reveal various kernel integrity/confidentiality manipulation and resource starvation attacks launched by compromised drivers, thus to assess the trustworthiness of the evaluated drivers.

System Call Redirection: A Practical Approach to Meeting Real-world Virtual Machine Introspection Needs

Existing VMI techniques have high overhead, and require customized introspection programs/tools for different guest OS versions – lack of generality. We developed ShadowContext [4], a system for close-to- realtime manual-effort-free VMI. ShadowContext can meet several important real-world VMI needs which existing VMI techniques cannot. Compared to other automatic introspection tool generation techniques, ShadowContext has two merits: (1) Its overhead is significantly less. It achieves close-to-realtime VMI. (2) It significantly improves the practical usefulness of introspection tools by allowing one introspection program to inspect a variety of guest OS versions. These merits are achieved via a new concept called "Shadow Context" which allows the guest OSes system call code to be reused inside a "shadowed" portion of the context of the out-of-guest inspection program. Besides, ShadowContext is secure enough to defend against a variety of real world attacks. ShadowContext is designed, implemented and systematically evaluated. Experimental results show that the performance overhead is about 75% with a median initialization time of 0.117 milliseconds.

Enabling Security-Aware Virtual Machine Placement in IaaS Clouds

Infrastructure as a Service (IaaS) facilitates the provisioning of virtual machines (VMs) in cloud computing platform for disjoint customers in a highly scalable, flexible, and cost-efficient fashion. However, provisioning of new VMs should take into account presence of vulnerable co-resident VM. A vulnerable VM poses security risk to co-locating VMs and the physical machine. Thus, VM placement policies can have an impact on the overall security of the cloud computing platform. In this work, we quantify the security risks of cloud environments for VM placement schemes in presence of vulnerable VMs. Based on our security evaluation, we propose a novel virtual machine placement scheme [1] that can minimize the security risks for the cloud platform.

Experimental results demonstrate that our approach can improve the survivability of most virtual machines and reduce the threat of attacks in the cloud platform. The computing costs and deployment costs of our techniques are also practical.

Effective resource allocation algorithm is critical to ensure the performance of users' applications as well as the efficiency of overall resource usage in cloud data center. We propose a new network-aware resource allocation solution based on minimum-height tree, with the goal of minimizing the maximum latency in communication between VMs as well as the overall network costs inside the data center. In our approach, we try to improve the effectiveness of resource allocation procedure by taking into account the hierarchical network topology characteristics of the data center. We also consider VM heterogeneities in terms of computational and communicational requirements to make our approach more practical. Simulations over exampled cloud systems imply that our approach can provide significant gains over other simpler resource allocation algorithms.

Detangling Resource Management Functions from the TCB in Cloud Virtual Machines

Recent research has developed virtualization architectures to protect the data privacy of guest virtual machines in cloud computing environments. The key technology is to include an access control matrix in the hypervisor. However, existing approaches have either limited functionalities in the hypervisor or a Trusted Computing Base (TCB) which is too large to secure. We proposed a new architecture, MyCloud SEP [9], to separate resource allocation and management from the hypervisor in order to reduce the TCB size while supporting privacy protection. In our design, the hypervisor checks all resource accesses against an access control matrix in the hypervisor. While providing flexibility of plugging-in resource management modules, the size of TCB is significantly reduced compared with commercial hypervisors. Using virtual disk manager as an example, we implement a prototype on x86 architecture. The performance evaluation results also show acceptable overheads.

Network-aware VM Migration

Host virtualization allows data centers to live migrate an entire Virtual Machine (VM) to support data center maintenance and workload balancing. Live VM Migration can consume nearly the entire bandwidth which impacts the performance of competing flows in the network. Knowing the cost of VM Migration propels data center admins to intelligently reserve minimum bandwidth required to ensure a network-aware VM migration. Recently, Remedy was proposed as a cost estimation model to calculate total traffic generated due to VM Migration.
Unlike the previous approaches, Remedy makes it possible to incorporate network topology leading to a more intelligent allocation of network resources during VM migration. However, Remedy was evaluated within a simulated environment running on a single machine. We empirically evaluated the performance of Remedy in an experimental GENI testbed characterized by wide-area network dynamics and realistic traffic scenarios [2, 5]. We deploy OpenFlow end to end QoS policies to reserve minimum bandwidths required for successful VM Migration. Preliminary results demonstrate that bandwidth reservation relieves the network of possible overloads during migration. We show that Remedy works best with link bandwidths of 1Gbps and above and pages dirty rates below 3000 pages/s. We present realistic scenarios that affect the accuracy of the cost estimation model. We conclude that link bandwidth, page dirty

rate and user specified progress amount are the critical parameters in determining VM migration cost.

Publications

1. Xuebiao Yuchi and Sachin Shetty**,** " Enabling Security-Aware Virtual Machine Placement in IaaS Clouds", Milcom, October 2015, Tampa FL.
2. Hellen Maziku and Sachin Shetty, "Network-Aware Virtual Machine Migration in Cloud Data Centers", 17th GENI Engineering Conference, Madison, Wisconsin, July 21- 23, 2013
3. Shengzhi Zhang, Peng Liu, "Assessing the Trustworthiness of Drivers," in Proceedings of The 15th International Symposium on Research in Attacks, Intrusions and Defenses (RAID '12), Amsterdam, Netherlands, September 12-14, 2012.
4. R. Wu, P. Chen, P. Liu, B. Mao, "System Call Redirection: A Practical Approach to Meeting Real-world VMI Needs," in Proceedings of the 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2014), Atlanta, Georgia, June 2014.
5. Hellen Maziku and Sachin Shetty, "Towards a Network Aware VM Migration: Evaluating the Cost of VM Migration in Cloud Data Center", IEEE CloudNet, October 2014, Luxembourg
6. Hellen Maziku and Sachin Shetty, "Network Aware VM Migration in Cloud Data Centers", Global Environment for Network Innovations (GENI) Research and Educational Experiment Workshop, March 19-20, 2014, Atlanta, GA
7. Lingchen Zhang, Sachin Shetty, Peng Liu and Jiwu Jing "RootkitDet: Practical End-to-End Defense against Kernel Rootkits in a Cloud Environment", European Symposium on Research in Computer Security, Wroclaw, Poland, September 2014
8. D. Tian, X. Xiong, C. Hu, P. Liu, "Defeating Buffer Overflow Attacks via Virtualization," Elsevier Journal on Computers & Electrical Engineering, accepted.
9. Min Li, Zili Zha, Wanyu Zang, Meng Yu, Peng Liu, Kun Bai. "Detangling Resource Management Functions from the TCB in Privacy-Preserving Virtualization." In The 19th European Symposium on Research in Computer Security (ESORICS 2014). September 7-11, 2014, Wroclaw, Poland.
10. Lingchen Zhang, Sachin Shetty, Peng Liu and Mohan Malkani, "Scalable Intrusion Detection System in the Cloud", Presented at 2013 AFOSR Information Operations and Security Meeting, Ballston, VA, August 5 - 9 2013